Journal of Intelligent Systems and Internet of Things
Vol. 12, No. 01, PP. 97-109, 2024
DOI: https://doi.org/10.54216/JISIoT.120108

Spider Monkey Optimization with Deep Learning-based Hindi Short Text Sentiment Analysis

Praloy Biswas[1,*], A. Daniel[2], Subhrendu Guha Neogi[3]

[1] Department of Computer Science and Engineering, Amity University Madhya Pradesh, Gwalior, India
[2] Department of Computer Science and Engineering, Amity University Madhya Pradesh, Gwalior, India
[3] Department of Computer Science and Engineering, The Neotia University, West Bengal, Kolkata, India
Emails: praloy.biswas.it@gmail.com; adaniel@gwa.amity.edu; sguhaneogi@gmail.com
[*]Corresponding Author: praloy.biswas.it@gmail.com

Abstract

Sentiment analysis (SA) intends to categorize a text respective to sentimental polarity of individual opinions, like neutral, positive, or negative. The study of Hindi is limited because of the grammatical and morphological complexities of the Hindi language while many research work concentrates on drawing features from English text. The hindi languages make the sentiment classification procedure for Hindi short text a tedious process. The Hindi language has complicated morphology and variation based on phonetics, spelling, and vocabulary; the common usage of numerous dialects between Hindi in India produces a massive volume of glossaries. In this study, we introduce a Spider Monkey Optimization with stacked recurrent neural network (SMO-SRNN) for short text SA on Hindi Corpus. The proposed SMO-SRNN technique mainly aims to identify and categorize the Hindi short text into three distinct classes, namely negative, positive, and neutral. In the presented SMO-SRNN method, the SRNN approach is exploited for the investigation and classification of sentiment. Moreover, the SMO model is employed to finetune the hyperparameter related to the SRNN model. A detailed set of experiments is applied to ensure the high efficiency of the SMO-SRNN algorithm. The comparative outcome highlighted the enhancement of the SMO-SRNN technique over other methods.

Received: August 16, 2023 Revised: November 09, 2023 Accepted: February 27, 2024

Keywords: Short text classification; sentiment analysis; deep learning; spider monkey optimization; recurrent neural network

1. Introduction

Hindi is the largely spoken branch of Semitic languages and the sixth official language in the United Nation. Many recent sentiment analysis studies (SA) can be carried out over English text; there were certain research works performed on Hindi text [1]. Thus, the technical sources produced, like natural language processing (NLP) tools and datasets, remain inadequate for Hindi. The nature of Hindi language will make the sentiment classifying procedure for short Hindi text a difficult task [2]. Different from the English language, Hindi orientation begin from right to left, and its letters are written with various shapes per their location in the word [3,4].

SA methods derive sentiments linked with particular subjects from online documents. Opinion Analysis is a task demanding a deep understanding of linguistics, text context, and domain knowledge [5-7]. Most NLP techniques utilized for SA need precise Part-of-Speech (PoS) tag data for a presented text [8]. This data can be offered by automated PoS tagging. Whereas PoS taggers morphological and analyzers for the Hindi language have been devised, such prevailing solutions do not support Dialect Hindi text, which can be seen on social media [9, 10].. Further, the lack of mass media text corpora which were large enough for training a tagging approach so that automated PoS tagging of mass media texts was hardly studied. Allowing the automated analysis of mass media content for extracting opinion intelligence and sentiment is very important to businesses and governments [12, 13]. Whereas business was concerned with learning people's opinions about their reputation, brands, and products, they were involved in the early detection of new trends (e.g., wellness trends, fashion, and sports,). This data is utilized for refining product offerings and advancing a competitive edge in the market. In contrast, governments display public opinion to understand prevalent views regarding current events and policies and detect extreme trends and views in public views that might indicate problematic situations [14].

This study presents a Spider Monkey Optimization with stacked recurrent neural network (SMO-SRNN) for short text SA on Hindi Corpus. The SMO-SRNN algorithm mainly identifies and categorizes the short Hindi text into three distinct classes: negative, positive, and neutral. In the presented SMO-SRNN method, the SRNN algorithm is employed to investigate and classify sentiments. Moreover, the SMO model is exploited to finetune the hyperparameters related to the SRNN technique. A detailed set of experiments is applied to ensure the high efficiency of the SMO-SRNN technique.

2. Literature Review

R Bhargava et al., (2019) discusses text sentiment analysis polarizes opinions. The study measures Hindi, Bengali, and Tamil twitter emotions. RNNs, LSTM, and CNN were utilized to create 39 sequential models with optimal parameter settings to reduce overfitting and error accumulation. The sequential models were tested in all three languages. Sequential models are examined to see how hidden layers affect approach performance. Neural networks were compared to normal machine learning to determine if they outperformed.

K Sarkar (2020) discusses social media communications like tweets and microblogging need sentiment analysis. Positive, negative, or neutral social media material is identified by polarity. This work detects Bengali and Hindi tweet sentiment polarity using heterogeneous classifiers in an ensemble. Our ensemble includes three basic classifiers with various feature sets. Researchers added emotion lexical polarity to tweets. Our heterogeneous ensemble model distinguishes Bengali and Hindi emotion polarity in experiments. Our Bengali and Hindi sentiment classifier is best [15].

S. Das et al., (2021) studies have employed social media sentiment analysis to explore human-computer interaction, consumer behavior, psychology, smart systems, etc. Sharing ideas on social media has boosted awareness of this issue due to its large data quantities. A tagged Hinglish dataset determines emotions. Transformer-based models and FastText multilingual word embeddings help deep learning discern emotions in Hindi-English tweets. Deep learning models CNN, LSTM, and Bi-LSTM assess sentiment. CNN was most accurate, 75.25%[16]. K Shanmugavadivel et al., (2022) discussed Naturally Language Processing analyzes text emotion. Emotions affect people and things. Text activities determine good, bad, or neutral. Evaluate statement influence on offensive language identification. Code-mixed data is used less for sentiment analysis and problematic language recognition than monolingual. Coded text is multilingual. Coded data conceals contempt. Monolingual data suits high-resource hostile language recognition and sentiment analysis. Machine learning, deep learning, and pre-trained BERT, RoBERTa, and adapter-BERT models should identify incorrect language and sentiment in low-resource code-mixed Tamil and English data. This study employs multitasking DravidianLangTech@ACL2022. By using word embedding to extract semantically meaningful code-mixed data. Offense language recognition (79%, sentiment 65%) lags adapter-BERT[17]. K Shrivastava (2020) studies blogs, retail websites, review portals, and social media, academics explore sentiment analysis. Hindi, Telugu, and Tamil grow quicker online than English, Chinese, and Spanish. Hindi online popularity and sentiment analysis are examined. Hindi e-newspaper movie reviews are examined for hidden emotions. Multiple-language reviews challenge sentiment analysis. This paper addresses deep learning issues with a Gated Recurrent Unit network and Hindi word embedding model. This approach classifies Hindi words by emotion using semantic and syntactic relationships. A genetic algorithm chooses the best Gated Recurrent Unit Network hyper-parameters. GA-GRU provides superior Hindi movie evaluations than resource-based and machine learning[18].

S.Das et al., (2023) discussed the popular in human-computer interaction, consumer behavior, psychology, and smart systems, social media sentiment analysis. The abundance of opinionated content on social media has enhanced awareness. The investigation will recognize emotions using Hinglish tags. Transformer-based models and FastText multilingual word embeddings extract emotions from mixed Hindi-English tweets using deep learning. CNN, LSTM, Bi-LSTM, and transformer models like BERT evaluate attitudes. CNNs outperform other models with 75.25% accuracy[19]. In [20], one of the word embedding approaches was utilised for training, like LSTM as a deep neural network (DNN) and an initial hidden layer for extracting features from input data. The method compiled with the Softmax layer was adopted for turning numerical outputs from the LSTM layer into probability for classifying the output as negative or positive.

3. The Proposed Model

This study introduces a new SMO-SRNN algorithm for short text SA on Hindi Corpus. The SMO-SRNN algorithm aims to detect and categorize the short Hindi text into 3 dissimilar classes: neutral, positive, and negative. In the presented SMO-SRNN technique, the SRNN model is employed to investigate and classify sentiments. Moreover, the

SMO method is exploited to fine-tune the hyperparameter related to the SRNN technique. The block diagram of the SMO-SRNN algorithm is shown in Fig. 1.

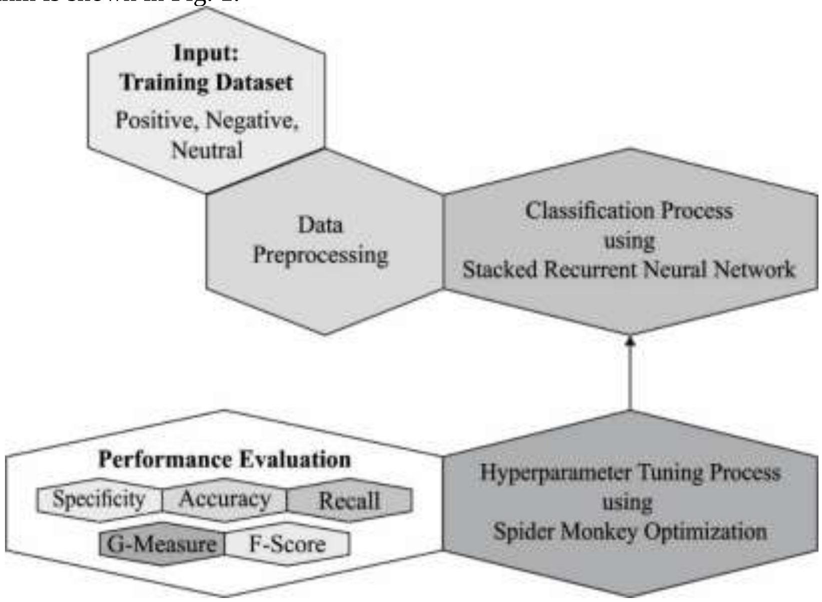

Figure 1: Block diagram of SMO-SRNN method

A. Stage I: Data Preprocessing

Hindi text pre-processing needs cleansing and normalizing techniques diverse from the techniques utilized for English and other languages. A word might comprise more than one possessive pronoun linked to morphemes. Thus, pre-processing stages for reducing word inflexion, like removal of stop words, stemming, and lemmatization, could not unavoidably have the similar effect since they were implemented in English text.

Bojanowski et al., 2017 presented another word embedding method named fastText. It depends on the unsupervised method for representing words as vectors. This is an expansion of Word2Vec method [22], from which the sub-words are considered. But fastText splits all the words into characters n-gram. It makes use of an angular bracket as a specific boundary as a sign of the starting and ending of the words. It is used to differentiate words and sub-words from other words. For example, the fastText presentation for the word "sentiment" if n = 4. Consider sub-words assist the presented method to differentiate among the suffixes and prefixes along with the short character sequence of the word. These models include the word to be characterized in a vector with the collection of n-gram characters. The sub-word is connected to the new words in the hashable list, and the amount of n-gram vectors is equivalent to the vector of a new word. Also, applied a similar dimension as in the preceding model, that is, 100, 200, and 300. As well we applied the minimum word count and similar window size to make a reasonable comparison.

B. Stage II: Sentiment Analysis Using SRNN Model

In the presented SMO-SRNN technique, the SRNN model is employed to investigate and classify sentiments. The study used RNN model for learning the feature representation of extremely imbalanced network traffic datasets for discriminatory classifiers [23]. The learning is performed X in batch as 3D tensors; thus, $X \in \mathbb{R}^{b \times t \times j}$, b represents the batch size, and t denotes the time step. Data with regard to formerly realized network traffic datasets are saved in hidden layer h. A tensors list can be applied for producing primarily hidden layer h_{init}. SRNN and RNN procedure present network traffic feature, x, cooperatively with the primary hidden layer, h_{init}, to generate a novel hidden layer, h_{1k}, shown below:

$$h_{1k} = \sigma_h \left(W_{xh} x_k + W_{hh} h_{init} + b_{\mathrm{h}}\right), \tag{1}$$

In Eq. (4), b_{h} denotes the bias vector; $W_{x\mathrm{h}}$ denotes the kernel weight matrixes applied for linear integration of input unit, x_k; σ_h indicates a Rectified Linear Unit (ReLU) and W_{hh} indicates the recurrent kernel weight matrixes applied for linear conversion of recurrent layer, h_{init}. A fully connected (FC) dense output layer has been applied for categorizing the output layer of RNN as follows:

$$\tilde{y}_k = \sigma_y (W_{\mathrm{h}v} h_{1k} + b_v), \tag{2}$$

In Eq. (2), $\tilde{y}_k$ denotes the forecasted label vector of SRNN; σ_y correspondingly indicates a $softmax$ or sigmoid function for multiple or binary class classifications; $W_{\mathrm{h}v}$ indicates the kernel weight matrixes applied for linear conversion of h_{1k}, and b_v denotes the bias vectors of the output layer. At last, we trained a RNN to implement multiple and binary-class classification tasks based on the Backpropagation Through Time (BPTT) approach.

$$W'_{(.)}\, b'_{(.)} = \psi(W_{(.)}, b_{(.)}), \tag{3}$$

In Eq. (3), ψ indicates the optimization function; $W_{(.)}$ and $W'_{(.)}$ denotes the older and newer Weight matrixes correspondingly; b and b show the older and newer bias vectors correspondingly. Different from the RNN with an individual hidden state, $(d-1)$ RNN layer is stacked through the initial RNN layer to implement hierarchical feature learning and increase classifier accuracy of extremely imbalanced datasets in SHN. This NN model has more than two RNNs, and is called SRNN. In the SRNN approach given in Algorithm 2, W_{xm} indicates the kernel weight matrixes utilized for linear conversion of $h_{(m-1)k}$; $W_{\text{h}m}$ shows the recurrent kernel weight matrixes utilized for linear conversion of the recurrent state, h_0; b_m indicates the bias vector of RNN layer, and m represents the amount of RNN layers in SRNN.

An FC dense output layer was applied to categorize the output of m^{th} RNN layer in SRNN as follows:

$$\tilde{y}_k = \sigma_y(W_{\text{h}v} h_{mk} + b_v), \tag{4}$$

In Eq. (4), $W_{\text{h}v}$ indicates the kernel weight matrixes utilized for his linear conversion. Eventually, SRNN has been trained to carry out binary and multi-class classification tasks based on the BPTT method.

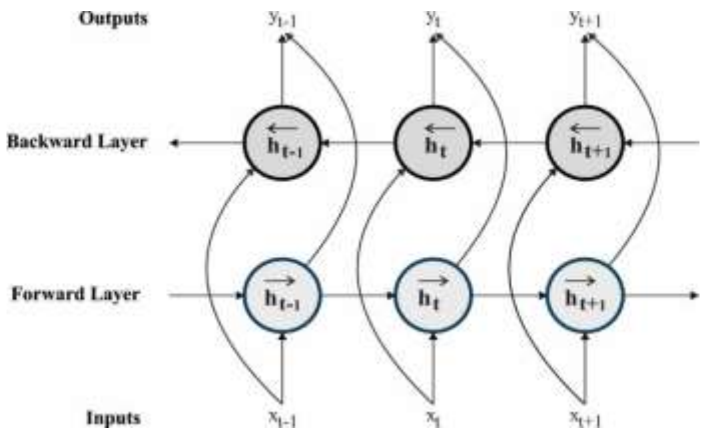

Figure 2: Structure of BiRNN

C. Stage III: Hyperparameter Tuning

Moreover, the SMO method is used for finetuning the hyperparameter related to the SRNN model. The traditional SMO algorithm is based on the spider monkey's intelligent behaviour. SMO and the variant are efficient and successful in addressing the complicated real-time optimization problem since they have higher efficacy [24]. SMO is a population based technique concerned with the social activities of spider monkey based on the clever hunting approach that replicates FFSS. In the FFSS, the individual creates a smaller cluster where member belongs to a stable community. They divide themselves into smaller clusters and are based conversely on obtainability and insufficient food sources. There exist six stages in this SMO technique, namely "LLP, GLP, LLL, GLL, LLD, and GLD, and it is discussed in the following.

Initializing population: in this phase, the population generation of M spider monkeys where Sma a=1,2 represent $s\ every\ monkey, \cdots, M$ refers to a D-dimension vector. And it is given as follows.

$$Sm_{ab} = Sm_{\min b} + rnd[0,1](Sm_{\max b} Sm_{\min b}) \tag{5}$$

Here, the limits of Sm_a in $the\ b^{th}$ direction are represented by $Sm_{\min b}$ and $Sm_{\max b}$. The arbitrary count is represented as, and that ranges from [0, 1].

LLP: here, the data collected fi: om the perspective of local leaders and members of local groups; thus, each monkey upgraded its corresponding location. The fitness value of the novel location is calculated, and change the location to the new one once the novel fitness is optimal. Eq. (6) represents the formula of location updating the spider monkey. Now, LLP_{cb} shows the b^{th} parameter of c^{th}leader position of the local groups. At the same time, Sm_{db} signifies the b^{th} parameter of $a\ d^{th}$ spider monkey randomly selected from $the\ c^{th}$ cluster so that $d \neq a$.

$$\begin{aligned} Sm_{newab} = Sm_{ab} + rnd[0,1](LLP_{cb} - Sm_{ab}) \\ + rnd[-1,1](Sm_{db} - Sm_{ab}) \end{aligned} \tag{6}$$

GLP: this procedure is done if LLP is accomplished. Here, each monkey re-evaluates their position by the memory of members of the local groups and the global leaders. The procedure of location updating can be implemented as follows.

$$\begin{aligned} Sm_{newab} = Sm_{ab} + rnd[0,1](GLP_b - Sm_{ab}) \\ + rnd[-1,1](Sm_{db} - Sm_{ab}) \end{aligned} \tag{7}$$

From the expression, the b^{th} parameter of the global leader location is provided as GLP_b, and $b \in \{1,2,\cdots,A\}$ indicates the randomly chosen index. Now, the location is upgraded according to the likelihood Pb_a that is described using the fitness values where the fitness of spider monkey is represented as fin_al, while the maximal fitness of group is shown as $ftn_{\max}$.

GLL: through greedy choice, the optimal location of spider monkey can be chosen based on the updating method. Then, the global leadership position is confirmed whether it is upgrading the location or not. Otherwise, the GLC is improved by one.

LLL: through the similar selection technique applied in GLL, the local leader location is chosen. In such cases, the novel position of the local leader corresponds to the older value or not. Once the location of the local leader was not upgraded, the LLC was increased by one.

LLD: in this phase, if the local leader location was not upgraded unless a pre-determined threshold called LLLimit, next, the member position presented in the smaller set is upgraded by a cooperative dataset from the global or local leaders otherwise randomly initialized through arithmetical formulated as follows.

$$Sm_{newab} = Sm_{ab} + rnd[0,1](GLP_b - Sm_{ab}) + rnd[-1,1](Sm_{db} - LLP_{cb}) \tag{8}$$

GLD: in this stage, the global leader location is confirmed that it can be upgraded or not till a pre-determined iteration count known as GLLimit; the population was divided into smaller sets via global leader. Once the maximal amount of sets is produced later, the location of a global leader isn't upgraded. In such cases, a small set is integrated into a unique set through the global leader.

4. Experimental Validation

A detailed set of experiments was applied on AraSenTi-Tweet dataset to inspect the SA outcomes of the SMO-SRNN method [25]. The dataset holds 15751 tweets with 4957 instances under the positive class, 4639 instances under the neutral class, and 6155 instances on negative class, as illustrated in Tab. 1

Table 1: Dataset details

Class	No. of Tweets
Positive	4957
Neutral	4639
Negative	6155
Total No. of Tweets	15751

The confusion matrix attained by the SMO-SRNN technique under several runs is exhibited in Fig. 3. On run-1, the SMO-SRNN method has detected 4881 instances in the positive class, 4503 instances in the neutral class, and 5970 instances in the negative class. Moreover, on run-2, the SMO-SRNN method has identified 4889 instances in the positive class, 4541 instances in the neutral class, and 6057 instances in the negative class. Further, on run-3, the SMO-SRNN method has identified 4880 instances in positive class, 4561 instances in the neutral class, and 6037 instances in the negative class. Finally, on run-4, the SMO-SRNN method has detected 4883 instances in positive class, 4543 instances in neutral class, and 6013instances in negative class. Finally, on run-5, the SMO-SRNN methodology has detected 4878 instances in the positive class, 4552 instances in neutral class, and 6058 instances in the negative class.

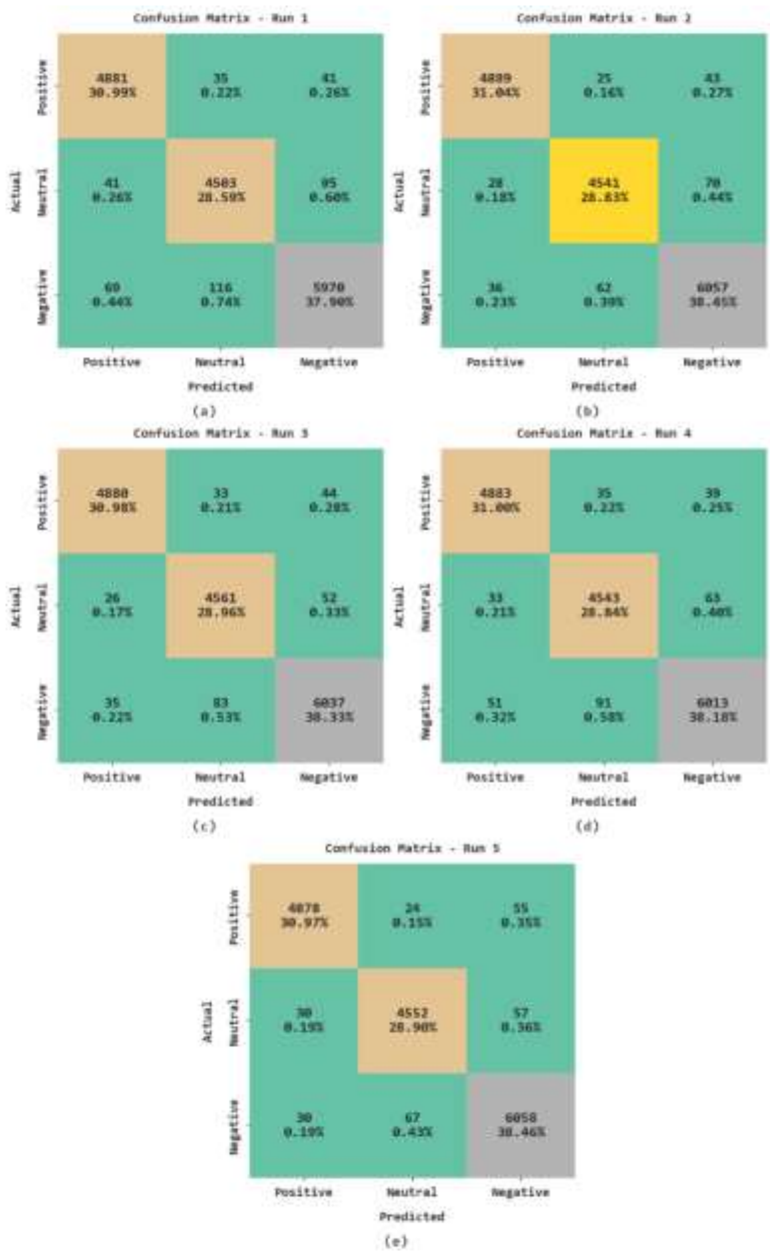

Figure 3: Confusion matrices of SMO-SRNN method (a) Run1, (b) Run2, (c) Run3, (d) Run4, and (e) Run5

The SA performance of the SMO-SRNN method on run-1 is demonstrated in Tab. 2 and Fig. 4. The obtained values indicated that the SMO-SRNN technique had recognized all the sentiments. The SMO-SRNN model has recognized positive tweets with $accu_y$, $reca_l$, $spec_y$, F_{score}, and $G_{measure}$ of 98.82%, 98.47%, 98.98%, 98.13%, and 98.13%, respectively. Additionally, The SMO-SRNN approach has recognized negative tweets with $accu_y$, $reca_l$, $spec_y$, F_{score}, and $G_{measure}$ of 98.18%, 97.07%, 98.64%, 96.91%, and 96.91%, correspondingly. Moreover, The SMO-SRNN approach has detected neutral tweets with $accu_y$, $reca_l$, $spec_y$, F_{score}, and $G_{measure}$ of 97.96%, 96.99%, 98.58%, 97.38%, and 97.38%, correspondingly.

Table 2: Result analysis of SMO-SRNN system with dissimilar classes on Run-1

Run-1					
Labels	$\boldsymbol{Accu_y}$	$\boldsymbol{Reca_l}$	$\boldsymbol{Spec_y}$	$\boldsymbol{F_{score}}$	$\boldsymbol{G_{measure}}$
Positive	98.82	98.47	98.98	98.13	98.13

Neutral	98.18	97.07	98.64	96.91	96.91
Negative	97.96	96.99	98.58	97.38	97.38
Average	98.32	97.51	98.73	97.47	97.48

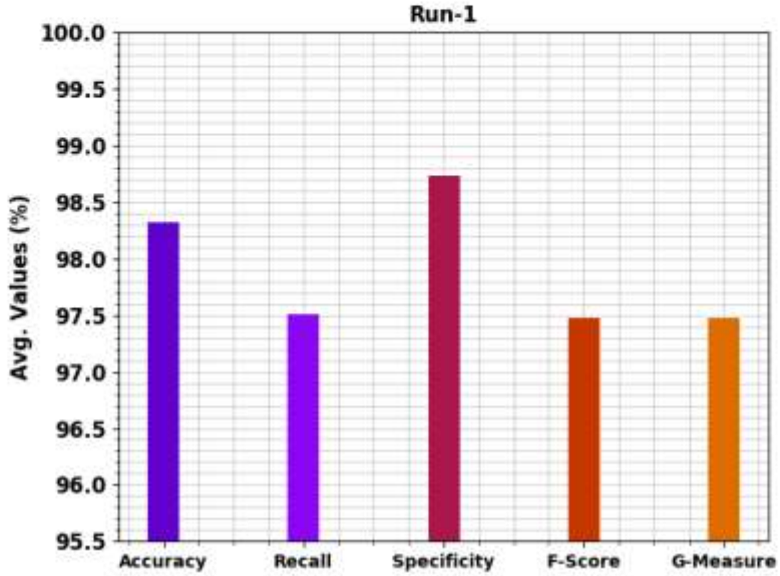

Figure 4: Result analysis of SMO-SRNN method on Run-1

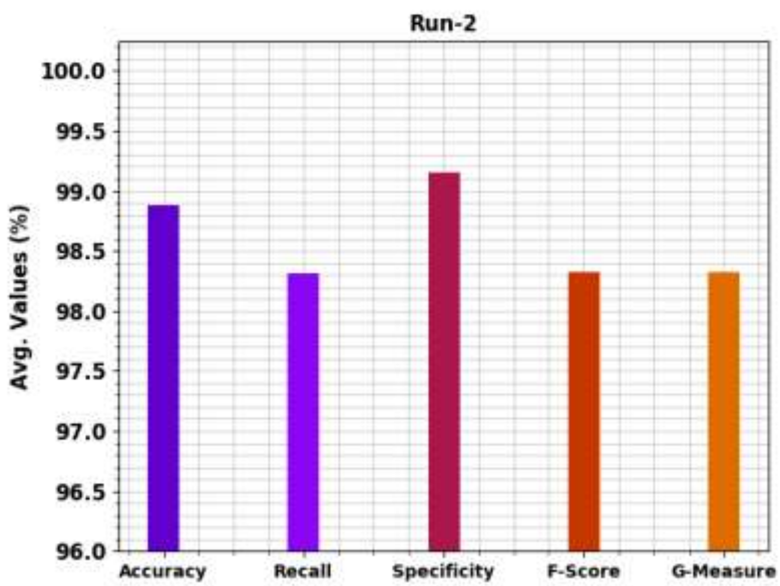

Figure 5: Result analysis of SMO-SRNN method on Run-2

Tab. 3 and Fig. 5 validate the SA outcomes of the SMO-SRNN model on run-2. The acquired values implied that the SMO-SRNN approach had recognized all the sentiments. The SMO-SRNN technique has recognized positive tweets with $accu_y$, $reca_l$, $spec_y$, F_{score}, and $G_{measure}$ of 99.16%, 98.63%, 99.41%, 98.67%, and 98.67% correspondingly. Additionally, The SMO-SRNN approach has recognized negative tweets with $accu_y$, $reca_l$, $spec_y$, F_{score}, and $G_{measure}$ of 98.83%, 97.89%, 99.22%, 98%, and 98%, respectively. Furthermore, The SMO-SRNN algorithm has recognized neutral tweets with $accu_y$, $reca_l$, $spec_y$, F_{score}, and $G_{measure}$ of 98.66%, 98.41%, 98.82%, 98.29%, and 98.29%, correspondingly.

Table 3: Result analysis of SMO-SRNN systems with different classes on Run-2

Run-2					
Labels	$Accu_y$	$Reca_l$	$Spec_y$	F_{score}	$G_{measure}$
Positive	99.16	98.63	99.41	98.67	98.67
Neutral	98.83	97.89	99.22	98.00	98.00
Negative	98.66	98.41	98.82	98.29	98.29
Average	98.88	98.31	99.15	98.32	98.32

Tab. 4 and Fig. 6 validate the SA performance of SMO-SRNN algorithm on run-3. The gained values implied that the SMO-SRNN approach had recognized all the sentiments. The SMO-SRNN technique has recognized positive tweets with $accu_y$, $reca_l$, $spec_y$, F_{score}, and $G_{measure}$ of 99.12%, 98.45%, 99.43%, 98.61%, and 98.61% correspondingly. Furthermore, The SMO-SRNN technique has recognized negative tweets with $accu_y$, $reca_l$, $spec_y$, F_{score}, and $G_{measure}$ of 98.77%, 98.32%, 98.96%, 97.92%, and 97.92%, correspondingly. Also, the SMO-SRNN algorithm has recognized neutral tweets with $accu_y$, $reca_l$, $spec_y$, F_{score}, and $G_{measure}$ of 98.64%, 98.08%, 99%, 98.26%, and 98.26% correspondingly.

Table 4: Result analysis of SMO-SRNN method with dissimilar classes on Run-3

Run-3					
Labels	$Accu_y$	$Reca_l$	$Spec_y$	F_{score}	$G_{measure}$
Positive	99.12	98.45	99.43	98.61	98.61
Neutral	98.77	98.32	98.96	97.92	97.92
Negative	98.64	98.08	99.00	98.26	98.26
Average	98.84	98.28	99.13	98.26	98.26

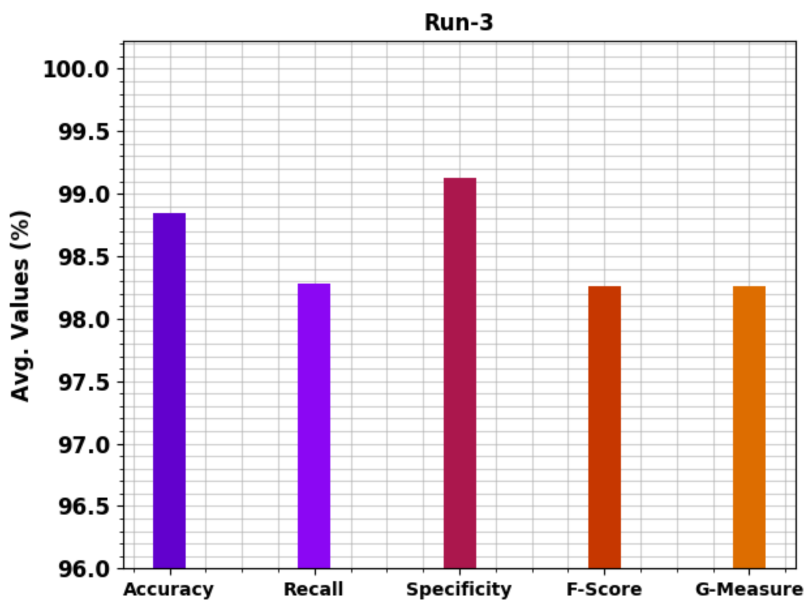

Figure 6: Result analysis of SMO-SRNN method on Run-3

Tab. 5 and Fig. 7 demonstrates the SA performance of the SMO-SRNN system on run-4. The acquired values implied that the SMO-SRNN approach has proficiently recognized all the sentiments. The SMO-SRNN methodology has recognized positive tweets with $accu_y$, $reca_l$, $spec_y$, F_{score}, and $G_{measure}$ of 99%, 98.51%, 99.22%, 98.41%, and 98.41% correspondingly. Moreover, The SMO-SRNN technique has recognized negative tweets with $accu_y$, $reca_l$, $spec_y$, F_{score}, and $G_{measure}$ of 98.59%, 97.93%, 98.87%, 97.61%, and 97.62%, respectively. Along with that, the SMO-SRNN approach has recognized neutral tweets with $accu_y$, $reca_l$, $spec_y$, F_{score}, and $G_{measure}$ of 98.45%, 97.69%, 98.94%, 98.01%, and 98.01% correspondingly.

Table 5: Result analysis of SMO-SRNN method with dissimilar classes on Run-4

Run-4					
Labels	$Accu_y$	$Reca_l$	$Spec_y$	F_{score}	$G_{measure}$
Positive	99.00	98.51	99.22	98.41	98.41
Neutral	98.59	97.93	98.87	97.61	97.62
Negative	98.45	97.69	98.94	98.01	98.01
Average	98.68	98.04	99.01	98.01	98.01

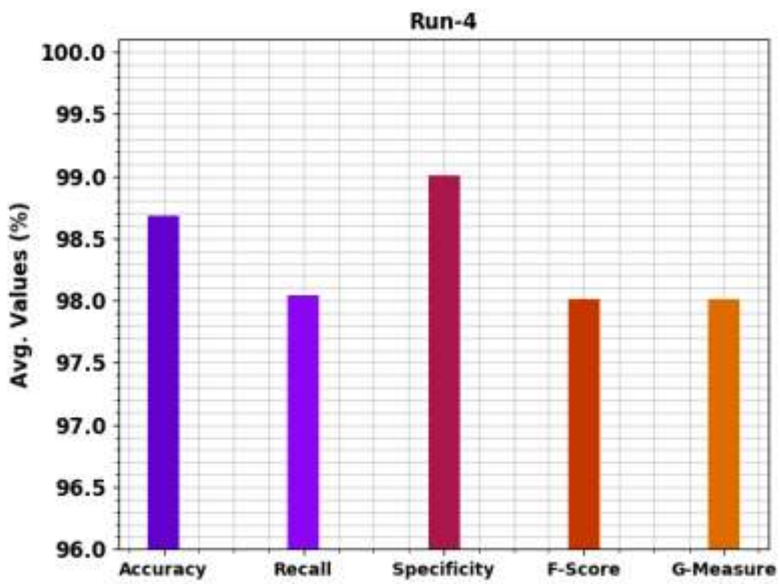

Figure 7: Result analysis of SMO-SRNN method on Run-4

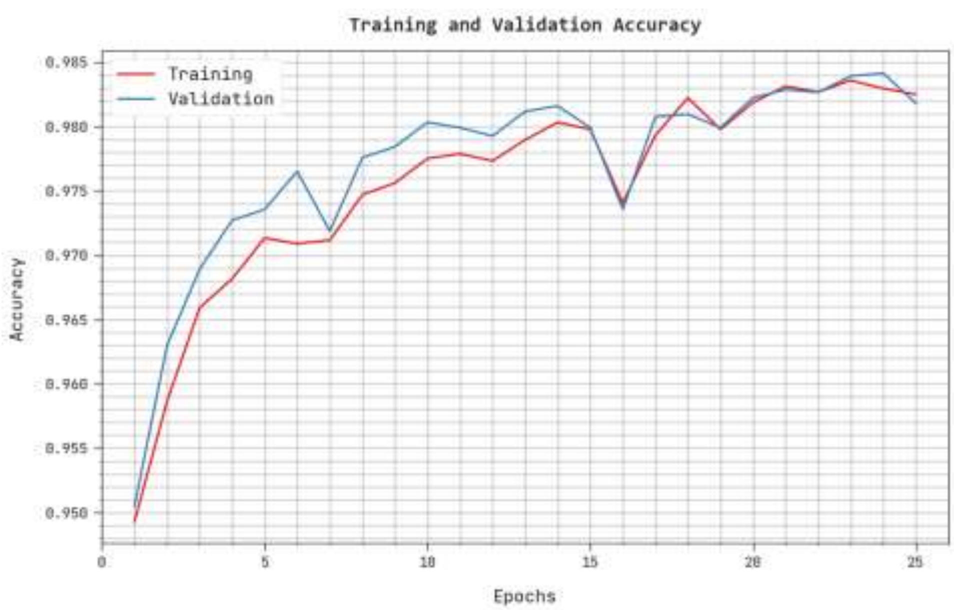

Figure 8: TRAC and VLAC analysis of SMO-SRNN method

The training accuracy (TRAC) and validation accuracy (VLAC) obtained by the SMO-SRNN methodology on the test dataset is displayed in Fig. 8. The outcomes indicated the SMO-SRNN approach has gained highest values of TRAC and VLAC. Especially, the VLA is superior to TRAC.

The training loss (TLOS) and validation loss (VLOS) reached by the SMO-SRNN method on test dataset are established in Fig. 9. The outcomes represents the SMO-SRNN technique has demonstrated minimum values of TLOS and VLOS. Specifically, the VLOS is lower than TLOS.

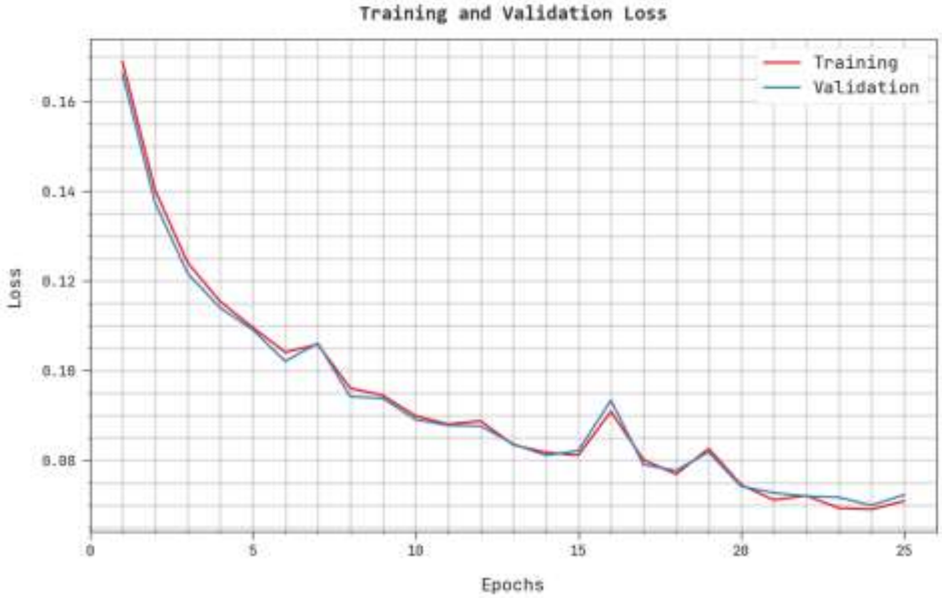

Figure 9: TLOS and VLOS analysis of SMO-SRNN method

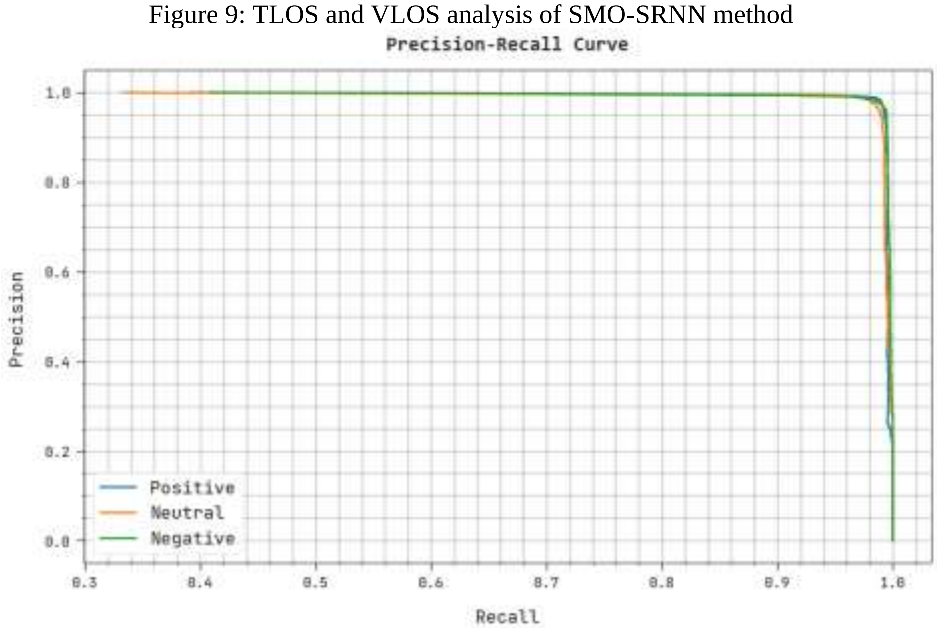

Figure 10: PR investigation of the SMO-SRNN methodology

A clear PR inspection of the SMO-SRNN system on the test dataset is demonstrated in Fig. 10. The figure signifies the SMO-SRNN approach has resulted in maximum PR values on all classes.
A brief ROC investigation of SMO-SRNN system on the test dataset is represented in Fig. 11. The outcomes mean the SMO-SRNN algorithm has shown its capacity to categorize dissimilar classes on the test dataset.

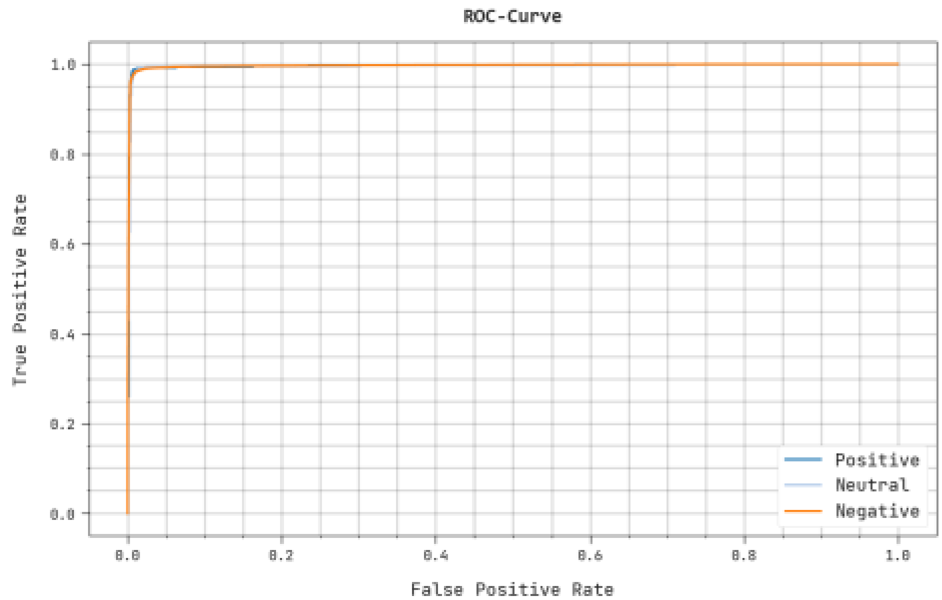

Figure 11: ROC analysis of SMO-SRNN method

Tab. 6 and Fig. 12 demonstrated the SA output of the SMO-SRNN method with compared methods [26]. Based on $accu_{racy}$, the SMO-SRNN model has demonstrated a higher $accu_y$ of 98.89%, whereas the LSTM, ANN, GNB, AlexNet, CNN, and DNN models have offered lower $accu_y$ of 94.44%, 94.30%, 92.76%, 97.71%, 94.53%, and 96.19% respectively. Along with that, based on rea_l, the SMO-SRNN approach has demonstrated a higher $reca_{ll}$ of 98.32%. In contrast, the LSTM, artificial neural network (ANN), Gaussian Naïve Bayes (GNB), AlexNet, CNN, and DNN models have granted lower $reca_l$ of 96.10%, 94.19%, 95.12%, 96.83%, 94.31%, and 96.80% correspondingly. At last, based on $spec$ ies, the SMO-SRNN method has demonstrated a higher $spec_y$ of 99.15%, whereas the LSTM, ANN, GNB, AlexNet, CNN, and DNN algorithms have rendered a lower $spec$ of 92.89%, 94.55%, 96.57%, 93.53%, 97.94%, and 96.00% correspondingly.

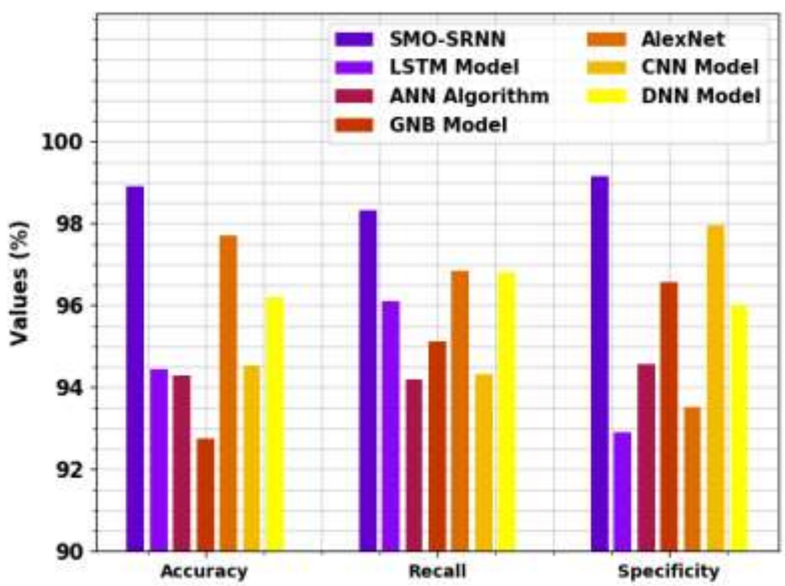

Figure 12: Comparative analysis of SMO-SRNN system with existing techniques

Table 6: Comparative analysis of SMO-SRNN technique with existing algorithms

Methods	$Accu_y$	$Reca_l$	$Spec_y$
SO-SRNN	98.89	98.32	99.15

LSTM Model	94.44	96.10	92.89
ANN Algorithm	94.30	94.19	94.55
GNB Model	92.76	95.12	96.57
AlexNet	97.71	96.83	93.53
CNN Model	94.53	94.31	97.94
DNN Model	96.19	96.80	96.00

From the results, it is obvious that the SMO-SRNN method has obtained superior outcomes over other existing SA techniques.

5. Conclusion

In this study, a new SMO-SRNN method was introduced for short text SA on Hindi Corpus. The SMO-SRNN algorithm aims to detect and categorize the short Hindi text into three dissimilar classes: negative, positive, and neutral. In the presented SMO-SRNN method, the SRNN model is employed to investigate and classify sentiments. Moreover, the SMO technique is used for finetuning the hyperparameter related to the SRNN technique. A detailed set of experiments is applied to ensure the high efficiency of the SMO-SRNN algorithm. The comparative outcomes highlighted the enhancement of SMO-SRNN technique over other methods. In the future, the presented SMO-SRNN method can be extended to SA in real-time environment.

References

[1] Sharma Y, Mangat V, Kaur M 2015 A practical approach to sentiment analysis of Hindi tweets. In: Proceedings of the 2015 1st International Conference on Next Generation Computing Technologies (NGCT), pp. 677–680.

[2] Joshi A, Balamurali A R, Bhattacharyya P 2010 A fall-back strategy for sentiment analysis in Hindi: a case study. In: Proceedings of the 8th ICON.

[3] Mittal N, Agarwal B, Chouhan G, Bania N, Pareek P 2013 Sentiment analysis of Hindi reviews based on negation and discourse relation. In: Proceedings of the 11th Workshop on Asian Language Resources.

[4] Sasidhar, T. T., et al. (2020). Emotion detection in hinglish (hindi+ english) code-mixed social media text. Procedia Computer Science, 171, 1346–1352.

[5] Ali, M. A., & Kulkarni, S. B. (2020). Preprocessing of text for emotion detection and sentiment analysis of Hindi movie reviews.

[6] Neogi, A. S., et al. (2021). Sentiment analysis and classification of indian farmers' protest using twitter data. International Journal of Information Management Data Insights, 1(2), 100019.

[7] Gupta, V., Jain, N., Shubham, S., Madan, A., Chaudhary, A., & Xin, Q. (2021). Toward integrated CNN-based sentiment analysis of tweets for scarce-resource language—Hindi. Transactions on Asian and Low-Resource Language Information Processing, 20(5), 1-23.

[8] Sharma, P., & Moh, T. S. (2016, December). Prediction of Indian election using sentiment analysis on Hindi Twitter. In 2016 IEEE international conference on big data (big data) (pp. 1966-1971). IEEE.

[9] Sai Aparna, T., Simran, K., Premjith, B., & Soman, K. P. (2021). Aspect-based sentiment analysis in Hindi: Comparison of machine/deep learning algorithms. In Inventive Computation and Information Technologies: Proceedings of ICICIT 2020 (pp. 81-91). Springer Singapore.

[10] Akhtar, M. S., Ekbal, A., & Bhattacharyya, P. (2016, May). Aspect based sentiment analysis in Hindi: resource creation and evaluation. In Proceedings of the tenth international conference on language resources and evaluation (LREC'16) (pp. 2703-2709).

[11] Shrivash, B. K., Verma, D. K., & Pandey, P. (2023). An Effective Framework for Sentiment Analysis of Hindi Sentiments Using Deep Learning Technique. Wireless Personal Communications, 132(3), 2097-2110.

[12] Joshi, R., Goel, P., & Joshi, R. (2020). Deep learning for hindi text classification: A comparison. In Intelligent Human Computer Interaction: 11th International Conference, IHCI 2019, Allahabad, India, December 12–14, 2019, Proceedings 11 (pp. 94-101). Springer International Publishing.

[13] Rani, S., & Kumar, P. (2019). Deep learning based sentiment analysis using convolution neural network. Arabian Journal for Science and Engineering, 44, 3305-3314.

[14] Sakshi Gupta, Piyush Bansal, and Radhika Mamidi. 2016. Resource creation for hindi-english code mixed social media text

[15] Bhargava, R., Arora, S., & Sharma, Y. (2019). Neural network-based architecture for sentiment analysis in Indian languages. Journal of Intelligent Systems, 28(3), 361-375.

[16] Sarkar, K. (2020). Heterogeneous classifier ensemble for sentiment analysis of Bengali and Hindi tweets. Sādhanā, 45(1), 196.

[17] Das, S., & Singh, T. (2023, January). Sentiment Recognition of Hinglish Code Mixed Data using Deep Learning Models based Approach. In 2023 13th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 265-269). IEEE.

[18] Shanmugavadivel, K., Sathishkumar, V. E., Raja, S., Lingaiah, T. B., Neelakandan, S., & Subramanian, M. (2022). Deep learning based sentiment analysis and offensive language identification on multilingual code-mixed data. Scientific Reports, 12(1), 21557.

[19] Shrivastava, K., & Kumar, S. (2020). A sentiment analysis system for the hindi language by integrating gated recurrent unit with genetic algorithm. Int. Arab J. Inf. Technol., 17(6), 954-964.

[20] Das, S., & Singh, T. (2023, January). Sentiment Recognition of Hinglish Code Mixed Data using Deep Learning Models based Approach. In 2023 13th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 265-269). IEEE.

[21] Jamatia, A., Swamy, S. D., Gambäck, B., Das, A., & Debbarma, S. (2020). Deep learning based sentiment analysis in a code-mixed English-Hindi and English-Bengali social media corpus. International journal on artificial intelligence tools, 29(05), 2050014..

[22] A. Onan, "Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks," Concurrency and Computation: Practice and Experience, vol. 33, no. 23, pp. 5909, 2021.

[23] Y. Tian, J. Zhang and J. Morris, "Modeling and optimal control of a batch polymerization reactor using a hybrid stacked recurrent neural network model," Industrial & Engineering Chemistry Research, vol. 40, no. 21, pp. 4525-4535, 2021.

[24] S. Kumar, B. Sharma, V. K. Sharma and R. C. Poonia, "Automated soil prediction using bag-of-features and chaotic spider monkey optimization algorithm," Evolutionary Intelligence, vol. 14, no. 2, pp. 293-304, 2021.

[25] Kulkarni, D. S., & Rodd, S. S. (2021). Sentiment Analysis in Hindi—A Survey on the State-of-the-art Techniques. Transactions on Asian and Low-Resource Language Information Processing, 21(1), 1-46

[26] Lamia Mohamed Ahmed , Gawaher Soliman Hussein, Abdel Nasser Hessin Zaied. "A Survey on Sentiment Analysis Algorithms and Techniques For Arabic Textual Data." Fusion: Practice and Applications, Vol. 2, No. 2, 2020 ,PP. 74-87 (Doi : https://doi.org/10.54216/FPA.020205)